SNISupport

SNI support can be enabled when running on Oracle JRE 7 by using a custom SSL socket factory. Please note this code snippet makes use of Commons BeanUtils to invoke Oracle private method via reflection.

HttpClient 4.3

```
SSLContext sslcontext = SSLContexts.createSystemDefault();
SSLSocketFactory sslsf = new SSLSocketFactory(sslcontext) {

    @Override
    public Socket connectSocket(
        int connectTimeout,
        Socket socket,
        HttpHost host,
        InetSocketAddress remoteAddress,
        InetSocketAddress localAddress,
        HttpContext context) throws IOException, ConnectTimeoutException {
        if (socket instanceof SSLSocket) {
            try {
                 PropertyUtils.setProperty(socket, "host", host.getHostName());
            } catch (NoSuchMethodException ex) {
            } catch (IllegalAccessException ex) {
            } catch (InvocationTargetException ex) {
            }
        }
        return super.connectSocket(connectTimeout, socket, host, remoteAddress,
            localAddress, context);
    }

};

CloseableHttpClient httpclient = HttpClients.custom()
    .setSSLSocketFactory(sslsf)
    .build();
CloseableHttpResponse response = httpclient.execute(new HttpGet("https://verisign.com/"));
try {
    System.out.println(response.getStatusLine());
    EntityUtils.consume(response.getEntity());
} finally {
    response.close();
}
```